openSUSE.org Build Service
Maintain one source for all Linux platforms

Putting cross development support into OBS

Martin Mohring
5e Datasoft GmbH

martin.mohring@opensuse.org
How to join such a FOSS project

• OBS is a corporate FOSS project, mostly Novell funded
• High entry level due to long SUSE history
• Reengineering of best practise and new technology

• How did I then join?
  - Pragmatic approach → fill a gap → Testing
  - Gain creditability
  - Gain write access to code
  - Provide packages for experimental users → New use cases

• Result:
  - Maintainer of Developer / Testing packages
  - Reengineered complete embedded knowhow inside OBS
  - Works
Types of Cross Development

Develop Software for another processor than the host

- **Type 1:** Build a complete set of packages using one big Package
  - e.g. busybox and buildroot
- **Type 2:** Put a complete distro including cross-tools into a Project
  - e.g. stlinux.com and OE
  - No original packages for the target can be used.
- **Type 3:** The build system is modified but use original source packages
  - No Examples outside OBS
- **Type 4:** Use emulation and/or virtualization and/or native build
  - e.g. scratchbox and mojo
  - original binaries can run
  - testsuites can run
Requirements I

• The goal was to put support for cross development into OBS as a generic, orthogonal feature
• Existing distributions should be usable as it is the case with x86 and ppc
• For the application developer, new targets should not be more than any other additional Linux target
• The codepath for normal operation (not cross-build) should behave exactly like without cross-build.
Requirements II

• Must work with server and with local build
• Normal OBS users without arrays of disks must be able to use it
• Existing meta data from .deb and .rpm FTP trees should be usable
• Split up in two sub-features:
  → Download on Demand
  → Cross Development
About Virtualization in OBS

• XEN/UML/QEMU/KVM can be used in workers
  – was in the first place a security feature
  – was a max compatibility feature on native Hardware

• For compatibility, system emulation was considered, but found too slow for cross development

→ Cross Development uses QEMU user emulation and Virtualization
Download on Demand I

• Big Distros need up to 20 GB / arch for the binary packages
• Remote OBS is only usable if one OBS stores all packages
• What about the FTP trees for all the distros?
• What if I want to use also the update FTP trees?
Download on Demand II

- Download on Demand caches only needed packages
- can read usual meta data from FTP trees:
  - debian md,
  - rpm md and
  - suse tags
- fire and forget (I did miss package xyz)
Implementation – the beginnings

• MicroSUSE

• Type 1: bring in uClibc buildroot
  - Put generic Infrastructure in place
  - Macro Processing works everywhere
  - Bring in notation of processor Architectures
  - Get powerpc working as first non x86 a

• Type 2: Import first Cross Build Distro stlinux.com

→ Now effort data present on amount of work for Type 1-4 to get a “real distro” building and running
Components Overview

- **Software search & installation Web client**
- **Command Line Client**
- **Web Client**
- **Installer (YaST, etc.)**

**Frontend**

- openSUSE API

**Backend**

- Build Host
- Build Host
- Build Host
- Build Host
- Build Host
- Build Host

**Storage**
Implementation – Changes in OBS

- OSC
- Webclient
- Frontend
- Scheduler
- Dispatcher
- Repo Server
- Worker
- Build
Implementation in the Backend

- Src server
- Rep server
- Publisher
- Repos
- Scheduler i586
- Dispatcher
- Workers i586
Status I

- First Release implemented
- Code supplied in the svn trunk since Dec/2008
- Packages provided for testing even since first versions
- Some Instances running – got feedback from there
- Implemented Metadata:
  - RPM
  - Deb
  - Susetags
Status II

- Type 1-4 Cross Development implemented
  - even combinations possible
- Compatibility paradigm proven
  - Maemo SDK implementation in 3 days
- Tested with lots of ARM Distros
- Ready to be supplied in public OBS for ARM
Testing Results I

Widely tested on Distros mostly for ARM

- Debian
- Ubuntu
- Fedora
- Maemo
- STLlinux (ARM9+11 + sh4) – Type 2
Testing Results II

• ARM processor levels
  – from armv4t (ARMv4 OABI) – Debian Etch
  – to armv7el (ARMv7 EABI + VFP) - Ubuntu/mojo

• Resultung packages installed on boards and run

• Most openSUSE Base system packages bootstrapped
  – inclusive running parts of the testsuites
Roadmap

- Activate ARM builds in public OBS
- Make Download on Demand more user friendly
- Implement optimizations
- Set up an ARM version of openSUSE
- Remove the bugs that pop up in public service
- Get also non ARM architectures running
  - mostly a QEMU user mode issue
- Support “non PC type” of Images

→ Lots of embedded Devices to assimilate
Resources

- http://build.opensuse.org
- A running instance of the Build Service.
- Contains links to documentation and source
- http://en.opensuse.org/Category:Build_Service
- Wiki documentation class for Build Service
- opensuse-buildservice@opensuse.de
- The mailing list for discussing the Build Service.
- #opensuse-buildservice on freenode
- Our IRC channel
- #opensuse-arm on freenode
- Our IRC channel for OBS and openSUSE @ ARM